

Όνομ/νο: Θανάσης Λώλος
AM: 1115201700072

ΟΔΗΓΙΕΣ ΧΡΗΣΗΣ

MAKEFILE

Με την εντολή “make [‘όνομα αρχείου.c’.o]” μπορούμε να παράγουμε ένα ένα τα object files (για separate compilation).

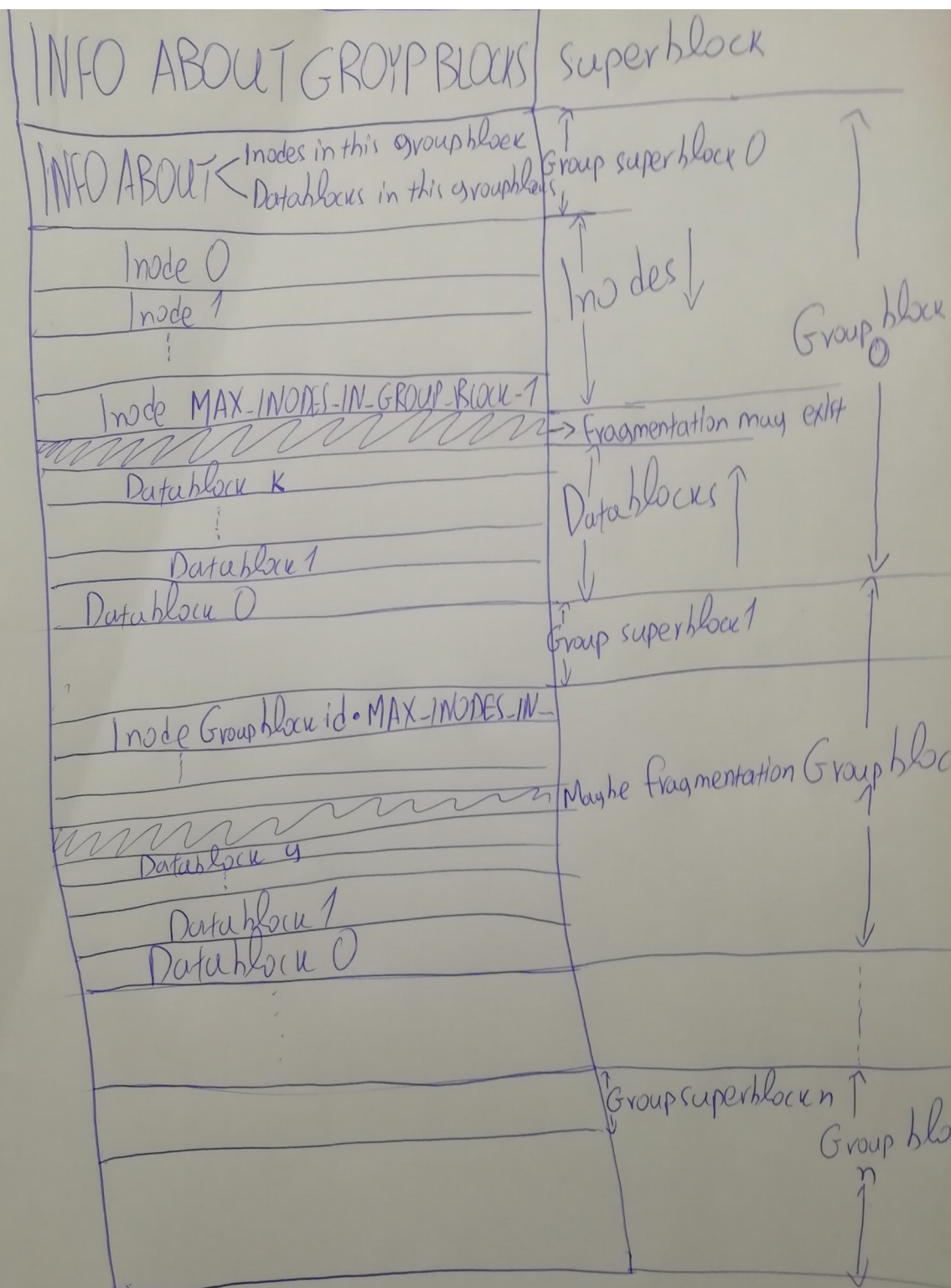
Με την εντολή “make” παράγεται το εκτελέσιμο ./file_system

```
gcc -c super_blocks/super_blocks.c -o super_blocks/super_blocks.o
gcc -c inode/inode.c -o inode/inode.o
gcc -c cursor/cursor.c -o cursor/cursor.o
gcc -c input_check/input_check.c -o input_check/input_check.o
gcc -c io/io.c -o io/io.o
gcc -c file_system.c -o file_system.o
gcc -c sort.c -o sort.o
gcc -c fs_operations/fs_operations.c -lm -o fs_operations/fs_operations.o
gcc super_blocks/super_blocks.o inode/inode.o cursor/cursor.o input_check/input_check.o io/io.o file_system.o sort.o fs_operations/fs_operations.o -o file_system
```

++make author → εμφανίζει τα στοιχεία μου

++make clean → σβήνει όλα τα object files

-----ΣΧΕΤΙΚΑ ΜΕ ΤΗΝ ΔΟΜΗ ΤΟΥ FILE SYSTEM-----
ΓΕΝΙΚΑ Η ΔΟΜΗ ΤΟΥ FS ΕΙΝΑΙ ΠΑΡΟΜΟΙΑ ΜΕ ΑΥΤΗΝ ΤΟΥ EXT2



- Στην αρχή του .cfs υπάρχει το SUPERBLOCK που περιέχει πληροφορίες σχετικά με το πόσα Group blocks υπάρχουν κλπ...
(Data block size, Inode struct size, Group block size, Max files in data block, Max filename size, Max inodes in block group, Max datablocks per file → -cfs)
- Στη συνέχεια επεκτείνονται τα Groupblocks

-----ΣΧΕΤΙΚΑ ΜΕ ΤΑ GROUP BLOCKS-----

-Στην αρχή περιέχουν ένα Group superblock.

Το group superblock περιέχει πληροφορίες σχετικά με τα inodes και τα data blocks που βρίσκονται στο συγκεκριμένο Group block.

Στη συνέχεια επεκτείνονται τα Inodes → ΜΕΓΑΛΩΝΟΥΝ ΠΡΟΣ ΤΑ ΚΑΤΩ
Απ'το τέλος του Group block ξεκινάνε τα Data block.

Τα Data block μεγαλώνουν δυναμικά προς τα πάνω → εως ότου συναντήσουν το τελευταίο Inode.(Λογική stack και heap σε ένα ELF).

-ΚΑΘΕ GROUP BLOCK ΕΧΕΙ ΕΝΑΝ MAX ΑΡΙΘΜΟ ΑΠΟ INODE ΠΟΥ ΜΠΟΡΕΙ ΝΑ ΦΙΛΟΞΕΝΗΣΕΙ.

→ $(GBLOCK_SZ - gsuperblock_sz) / (\text{Inode struct size} + \text{One datablock})$.

-ΕΝΑ GROUP BLOCK ΜΠΟΡΕΙ ΝΑ ΑΠΟΤΕΛΕΙΤΑΙ ΟΛΟΚΛΗΡΟ ΑΠΟ DATA BLOCKS.

Για παράδειγμα έστω ότι ένα group block έχει Max inodes in block
group=25

Το Group block 0 μπορεί να περιέχει : Inodes num (0-24)

Το Group block 1 μπορεί να περιέχει : Inodes num (25-49)

....

Το Group block n μπορεί να περιέχει: $(n * \text{Maxinodes} - (n+1) * \text{Maxinodes})$

Έτσι μπορώ να προσπελάσω γραμμικά ($O(1)$) οποίο inode θέλω.

***Για να υλοποιηθεί το παραπάνω θεωρώ ότι *inodes numbers dedicated* σε κάθε Group block!**

Για παράδειγμα αν έχω σε κάποιο Group block ένα πολύ μεγάλο αρχείο μπορεί να μην μπορούν να μπουν σ'αυτό άλλα inodes.(Τα Datablocks του έχουν μεγαλώσει μέχρι πάνω-πάνω και δεν μένει χώρος που να χωράει άλλα)

Κάτι τέτοιο δεν αποτελεί πρόβλημα καθώς κάνω append καινούριο group block και μπορώ να φιλοξενήσω κι άλλα inodes που θα έχουν ids με αυτά που ταιριάζουν στα dedicated ids του εκάστοτε groupblock.

-cfs: Υποδεικνύει το max αριθμό data blocks μεγέθους -bs που έχει στη διάθεση του κάθε entity του fs (dir,regular file).

Το -cfs καθορίζει κατά κύριο λόγο το μέγεθος του inode struct.

Στα data του inode struct κρατάω μια δυναμική δομή που μου λέει που βρίσκονται τα data block του συγκεκριμένου entity.

Π.χ έστω inode file;

file.data.block → το array

file.data.size → είναι το μέγεθος του array με max size -cfs

file.data.block[0] → 1ο block

file.data.block[1] → 2ο block κλπ....

file.data.block[i].block_group → σε ποιο block group

file.data.block[i].block → ποιο data block του παραπάνω block group

Τα data blocks του κάθε group block ξεκινάνε απτο 0!

Με αυτόν τον τρόπο δεν χρειάζεται όλα τα data block ενός entity να βρίσκονται στο ίδιο group block, καθώς μπορεί να βρίσκονται διάσπαρτα σε άλλα group blocks → ή όπου γενικότερα υπάρχει χώρος.

-----ΣΧΕΤΙΚΑ ΜΕ ΤΑ DATA BLOCKS-----

Data blocks directory

Στην αρχή κάθε data block περιέχεται ο αριθμός των entries που είναι σ'αυτο το data block (sizeof(int)).

Στη συνέχεια επεκτείνονται τα entries.

Παράδειγμα datablock

25

entity1 inode_id_of_file1

entity2 inode_id_of_file2

....

entity25 inode_id_of_file25

δηλαδή: string inode_num

μεγέθους: MAX_FILENAME_SIZE sizeof(int)

Έτσι τα size των directories καθορίζονται με αυτόν τον τρόπο.

Έστω Max filename size: 16 τότε dir με 1 entity μέσα του έχει size 24

αντίστοιχα ένα dir με 2 entities μέσα του έχει size 44 κ.ο.κ

Data block regular file

Αντίστοιχα περιέχουν έναν int στην αρχή και στην συνέχεια επεκτείνονται τα δεδομένα του. → Ο int υποδεικνύει πόσα bytes υπάρχουν στη συνέχεια Έτσι για παράδειγμα αν έχω Data block size 512 τότε χωράνε σ'αυτο και σε κάθε άλλο 509 bytes.

-----ΣΧΕΤΙΚΑ ΜΕ ΤΑ HOLES-----

Holes μπορούν να δημιουργηθούν από την cfs_rm, αλλά και από την cfs_mv!

Απτήν cfs_mv δημιουργούνται καθώς όταν βγάλω ένα entry από ένα datablock ενός dir δημιουργείται hole στο ίδιο το datablock το οποίο γεμίζω με το τελευταίο entry του τελευταίου data block του ίδιου directory.

Εφόσον όμως πειράζω το size του τελευταίου datablock μπορεί να του πάρω το μοναδικό (και τελευταίο entry) και έτσι χρειάζεται να αποδεσμεύσω το συγκεκριμένο block καθώς πλέον δεν χρησιμοποιείται και είναι άδριο.

Γενικά τα datablock παρέχονται on-demand όταν ζητηθούν!

Ομοίως ισχύει και για τα inode ids.

ΛΟΓΙΚΗ ΠΑΡΟΧΗΣ DATA BLOCK

-δες αν χωράει data block στην κορυφή stack των datablock στο group block στο οποίο υπάρχει η εγγραφή inode

-Αν δεν υπάρχει ψάξε για κάποιο hole στο group block αυτό
ένα hole εντοπίζεται αν στην αρχή του ο int ισούται με -1 → υποδεικνύει κενό data block

-Αν τίποτα απτα παραπάνω δεν δούλεψε πήγαινε στο τελευταίο group block και επανέλαβε τα παραπάνω

-Αν πάλι τίποτα → append a new group block και δώσε το datablock 0

Η ΛΟΓΙΚΗ ΠΑΡΟΧΗΣ ΔΙΑΘΕΣΙΜΟΥ INODE ID ΕΙΝΑΙ ΙΔΙΑ ΜΕ ΤΗΝ ΠΑΡΑΠΑΝΩ. → αντί για -1 στην αρχή ενός inode στο .cfs αναγράφεται το string EMPTY!

ΜΕ ΤΙΣ ΠΑΡΑΠΑΝΩ ΛΟΓΙΚΕΣ ΤΑ HOLES ΚΑΠΟΙΑ ΣΤΙΓΜΗ ΘΑ ΧΡΗΣΙΜΟΠΟΙΗΘΟΥΝ.

ΣΤΟ FILE SYSTEM ΔΕΝ ΕΧΟΥΝ ΥΛΟΠΟΙΗΘΕΙ LINKS

ΣΥΝΑΡΤΗΣΕΙΣ

Δεν έχει υλοποιηθεί η cfs_ln

cfs_cp: ΟΧΙ -R μόνο -r όπως έγραψε κάποιος στο piazza(Andrew Spanopoulos)

cfs_ls: όχι πολλά files

cfs_rm : ΟΧΙ -r → Γίνεται delete ότι επιλεγεί από -i αν δεν υπάρχει -i σβήνεται ό,τι δωθεί.

cfs_create: -cfs υποδεικνύει max αριθμό blocks κάθε entity.

ΟΧΙ -mdfn καθώς αυτό καθορίζεται από -cfs

ΥΠΟΣΤΗΡΙΖΟΝΤΑΙ ABSOLUTE ΚΑΙ RELATIVE PATH ΓΙΑ ΟΛΕΣ ΤΙΣ ΕΝΤΟΛΕΣ ΕΚΤΟΣ ΑΠΟ cfs_touch και cfs_mkdir!

Υλοποιήθηκε σε lubuntu.